Finding Malicious Cyber Discussions in Social Media *

Richard P. Lippmann, David J. Weller-Fahy, Alyssa C. Mensch,
William M. Campbell, and Joseph P. Campbell
MIT Lincoln Laboratory, 244 Wood Street, Lexington, MA 02420
{lippmann,djwf,alyssa.mensch,wcampbell,jpc}@ll.mit.ed

Abstract

Security analysts gather essential information on cyber attacks, exploits, vulnerabilities, and victims by manually searching social media sites. This effort can be dramatically reduced using natural language machine learning techniques. Using a new English text corpus containing more than 250k discussions from Stack Exchange, Reddit, and Twitter on cyber and non-cyber topics, we demonstrate the ability to detect more than 90% of the cyber discussions with fewer than 1% false alarms. If an original searched document corpus includes only 5% cyber documents, then our processing provides an enriched corpus for analysts where 83% to 95% of the documents are on cyber topics. Good performance was obtained using TF-IDF features and logistic regression. A classifier trained using prior historical data accurately detected 86% of emergent Heartbleed discussions and retrospective experiments demonstrate that classifier performance remains stable up to a year without retraining.

1 Introduction

Four essential tasks must be performed to secure any enterprise computer network. System administrators must (1) anticipate future attacks, (2) defend against attacks that can be prevented, (3) observe their own network to detect evidence of attempted and successful attacks, and (4) recover from successful attacks. Improving the ability to anticipate attacks improves defenses, targeting of observations, and the ability to recover from attacks. Anticipation can take advantage of social media conversations by criminal hackers and security researchers who discuss cyber attacks, exploits, vulnerabilities, strategies, tools, and actual and potential victims on the Internet. Cyber analysts currently manually examine the Internet to find these discussions.

Manual searches often use metadata (e.g., account names, thread or discussion topics, sources and destinations of social media discussions). This process is labor intensive and sometimes ineffective because attackers can easily hide malicious conversations in discussions on non-cyber topics (e.g. music or astronomy) and security researchers can post information using news feeds or forums that only contain a small percentage of cyber content. A more efficient and effective approach we explore in this paper is to supplement metadata analysis with direct mining of discussion text using machine-learning and human language technology (HLT) approaches.

The remainder of this paper is organized as follows. In Section 2, we assess related work in the application of HLT to the cyber domain. In Section 3, use-cases, text corpora, features, and classifiers are described. In Section 4, experiments and performance results are presented. Finally, in Section 5 we describe conclusions drawn from this work and future directions.

2 Related Work

There has been substantial research on HLT (e.g. [Jurafsky and Martin, 2008]) and cybersecurity (e.g.[Anderson, 2008]), but only a few researchers are beginning to apply HLT to the cybersecurity domain. Recent research [Sabottke *et al.*, 2015] demonstrated that finding keywords related to exploits and vulnerabilities across more than 250k Twitter tweets [Twitter, 2016] provides some useful evidence that a vulnerability listed in the National Vulnerability Database (NVD) [NIST, 2017] base will actually be exploited. In another recent paper [Lau *et al.*, 2014] interactions between known cyber criminals on social media were analyzed to distinguish between transactional interactions, in which cyber-attack tools are bought or sold, and collaborative interactions, in which cyber criminals share tools or information without any monetary exchange [Lau *et al.*, 2014]. Their analysis, does not identify cyber discussions, but requires manual extraction of cyber discussions before automated transaction analysis can be performed. Two recent papers focus on the problem of extracting relational information concerning vulnerabilities from text [Jones *et al.*, 2015; Syed *et al.*, 2016]. Example relations might be that "Adobe is_vendor_of Acrobat" or "CVE-2002-2435 is_a_vulnerability_in Acrobat". The exploratory study in [Jones *et al.*, 2015] only extracted vulnerability relations from 62 news articles and [Syed *et al.*, 2016] focused only on

*Distribution A: Public Release. This work is sponsored by the Department of Defense under Air Force contract FA8702-15-D-0001. Opinions, interpretations, conclusions, and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Corpus	Topics		Documents		Time Covered	Document Labeling Method
	Cyber	Non-cyber	Number of Documents	Median Number of Words		
Stack Exchange	5	10	~200k	245	Years	Community topic and tags
Reddit	10	51	~59k	152	Months (Non-cyber) Years (Cyber)	Sub-Reddit topic
Twitter	127	500	627	546	Months	Expert cyber users' tweets

Table 1: Social Media Corpora Document Labeling

well-structured NVD vulnerability descriptions, but mapped extracted entities to corresponding entities in the large DBpedia structured database [Auer *et al.*, 2007]. Such an extended database could be used to support queries concerning vulnerabilities and assist in extraction vulnerability information from other unstructured text. The more recent work in [Syed *et al.*, 2016] extended earlier work described in [Mulwad *et al.*, 2011] where entities such as software products, vulnerabilities, and organizations were extracted from a small set of 107 NVD vulnerability text descriptions. The conceptual approach described in [Mulwad *et al.*, 2011] includes a classifier to identify documents that describe vulnerabilities, but the simple text classifier described was developed and tested using only 155 example documents including only 75 NVD text descriptions as examples of documents describing vulnerabilities.

3 Classifier Development

Our major goal is to develop HLT classifiers that analyze the large number of discussions in online forums and discover the few that are most likely are concerned with attack methods, exploits, vulnerabilities, strategies, attack tools, defenses, and actual and potential victims that are of interest to cyber analysts. There are two potential use cases for a filter that detects malicious cyber topics. First, already discovered Internet content, such discussions under specified Reddit [Reddit, 2016] topics or lists of users in Twitter, can be processed to determine which content is most relevant. This ranking is necessary because discussions often drift off-topic and may move to other forums. Second, new Internet forums (e.g., Twitter hashtag threads) can be discovered that contain cyber discussions of interest. This scenario is more difficult because the search can extend across many types of social media and forums. It also would benefit from a classifier that works well across different forum and language types.

3.1 Training and Testing Corpora

Training requires both cyber and non-cyber social media discussions. Cyber discussions should be representative of those that are of most interest to analysts. Non-cyber training examples should contain discussions that cover many non-cyber topics to provide good performance during use when any of a wide range of non-cyber topics might be encountered. After a classifier is trained, it can be fed input text from a social media discussion and provide as output the probability that the discussion is on a cyber topic. An output probability supports both use-cases described above. Conversations in forums of interest can be ranked by probability, or many new forums can be scanned to identify those with the greatest number of probable cyber conversations.

We trained and tested our classifiers using text discussions from three social media forums: Stack Exchange [StackExchange, 2016], Reddit [Reddit, 2016], and Twitter [Twitter, 2016]. Stack Exchange is a well-moderated question-and-answer network with communities dedicated to many diverse topics. Questions and answers can be quite comprehensive, long, and well written. Reddit is a minimally moderated set of forums with main topics called sub-Reddit and many individual threads or discussions under each topic. Twitter data consist of short text messages (*tweets*) with at most 140 characters each. Tweets can be followed via user names and hashtags that identify tweets on a similar topic, or Twitter lists (i.e., curated groups of Twitter users). These three corpora were selected because they contain text with at least some cyber content, span a range of social media types, and offer a history of posts over a long time span.

Table 1 shows the number of cyber and non-cyber topics in each corpus, the number of documents and median number of words in documents, the time period covered by the collection, and a summary of how documents were labeled as cyber or non-cyber. Documents were created using all posts concerning discussions on a specific question for StackExchange, all posts for a specific sub-Reddit thread in Reddit, and collected tweets from users with a minimum of 20 and a maximum of 300 tweets. Preprocessing eliminated side information such as the date, thread title, hashtag, and user name that might not be available for future pure text searches. Documents for Stack Exchange and Reddit were labeled used topic titles and tags already provided in each corpus. All posts under cyber-related topics (e.g., reverse engineering, security, malware, blackhat) were labeled as cyber and posts on non-cyber topics (e.g., astronomy, electronics, beer, biology, music, movies, fitness) were labeled as non-cyber. For Stack Exchange we also used lower-level tags such as penetration-test, buffer-overflow, denial-of-service, and heartbleed to further identify cyber discussions. For Twitter, 127 cyber experts were identified and tweets from those experts were labeled

cyber while tweets from 500 other randomly selected users were labeled non-cyber. The disparity between the time covered for the cyber and non-cyber Reddit posts is a function of the limits imposed by the Reddit API: the most recent 999 posts in each sub-Reddit were collected and the non-cyber posts occurred at a much higher rate than the cyber posts.

3.2 Reference Keyword Detector

As a baseline reference to compare more advanced classifiers, we used a keyword-based approach that had been developed by security analysts to detect cyber discussions. This approach searches for 200 keywords and key phrases in documents and counts the number of occurrences. Higher counts indicate documents that are more likely about cyber topics. Examples of keywords include "rootkit," "Infected," and "checksum." Examples of phrases are "buffer overflow", "privilege escalation," and "Distributed Denial of Service." This reference keyword detector was similar to the approach used in [Sabottke *et al.*, 2015] to detect Twitter tweets discussing vulnerabilities except security analysts selected keywords by hand based on past experience.

3.3 Processing and Classification

Our classification pipeline requires preprocessing each document, generating an input feature for each distinct word in a document, and training a classifier to distinguish between cyber and non-cyber documents on the basis of these generated features. Preprocessing employs stemming to normalize word endings and text normalization techniques, such as the removal of words containing numbers and the replacement of URLs with a special token indicating a URL, to ensure that the feature inputs are standardized. We used term frequency - inverse document frequency (TF-IDF) features, created by counting the occurrences of words in documents and normalizing by the number of documents in which the words occur (e.g. [Jurafsky and Martin, 2008]). In our research and in the HLT community's research in general, TF-IDF features have provided good performance when used in text classification. For testing, the inverse document frequency counts were set as calculated during training. Most of the experiments used single word counts to create features. Using N-grams and feature selection provided only small differences in performance. We initial explored logistic regression classifiers (e.g. [Hastie *et al.*, 2009]), support vector machine (SVM) classifiers (e.g. [Cortes and Vapnik, 1995]) and Latent Dirichlet Allocation [Blei *et al.*, 2003] word features. All three approaches provided similar performance, required little computation to analyze a document, and provided an output proportional to the probability that an input is a cyber document. In the remainder of this paper, we describe results obtained using an L_2regularized logistic regression classifier for the Stack Exchange and Reddit corpora and using a linear SVM classifier for the smaller Twitter corpus.

4 Results

All results shown were obtained using 10-fold cross validation training/testing performed separately on each corpus. Figure 1 shows results for logistic regression classifiers on Stack Exchange [StackExchange, 2016], Twitter, and Reddit. Results for Stack Exchange are for all documents, results for Twitter are for documents for users with more than 100 tweets, and results for Reddit are for documents containing more than 200 words. Each classifier outputs the probability that a document discusses cyber topics. This probability can be used to label the document as cyber or non-cyber based on a set threshold (the minimum probability required for the classifier to label a document as cyber). The document labels then make it possible to determine the number of false alarms (i.e., non-cyber documents that are classified as cyber) and misses (i.e., cyber documents that are classified as non-cyber) for each threshold. We present our results in the form of detection error tradeoff (DET) curves that show false alarm and miss probabilities as the threshold on the classifier's output probability varies plotted on normal deviate scales [Martin *et al.*, 1997]. These curves make it possible to determine the output percentage of cyber documents for any input percentage of cyber documents and any threshold setting.

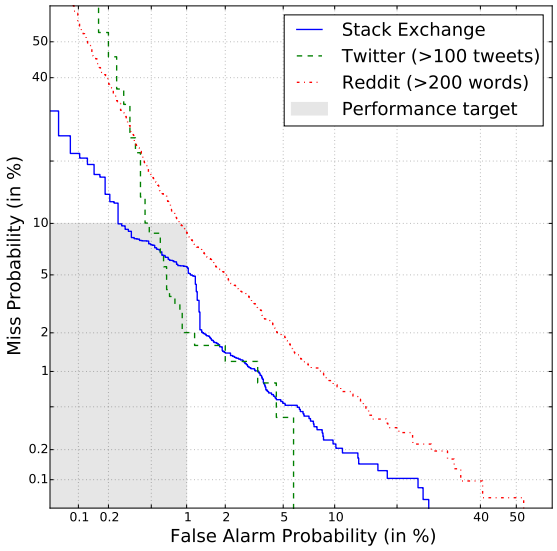

Figure 1: Detection error tradeoff plots indicate that the classifiers perform well for all three social media corpora missing fewer than 10% of input cyber documents at 1% false alarms.

Good performance in this figure is indicated by curves that are lower and closer to the bottom left. As shown by the gray box in Figure 1, a false alarm rate below 1% and a miss rate below 10% was selected as our performance target after discussions with analysts. This provides a high concentration of cyber documents in the filtered documents sent to analysts. If we assume that the initial set of documents to be classified has been pre-selected to contain 5% or 5 in 100 cyber documents, then a classifier with lower than 1% false alarms and lower than 10% misses will result in 83% or 83/100 cyber documents in the filtered stream of documents classified as

Top 50 Cyber Words	HTTP, SQL, Secur, URL, Window, access, address, app, application, attack, authenticate, browser, bug, certificate, client, code, crack, detect, encrypt, execute, exploit, file, firewall, hash, infect, inject, install, key, malicious, malware, network, obfuscate, overflow, packet, password, payload, request, risk, scan, script, secure, server, site, test, tool, traffic, user, virus, vulnerability, web
Top 50 Non-cyber Words	Arduino, Christian, God, LED, The, and, bank, board, buy, cell, chip, chord, circuit, clock, credit, current, datasheet, design, electron, film, frac, frequency, fund, graph, hi, invest, microcontroller, motor, movie, music, note, output, part, pin, play, power, rate, resistor, serial, signal, simulate, state, stock, tax, the, time, tree, two, voltage, wire

Table 2: The cyber and non-cyber words with the largest magnitude coefficients in our classifier trained on Stack Exchange.

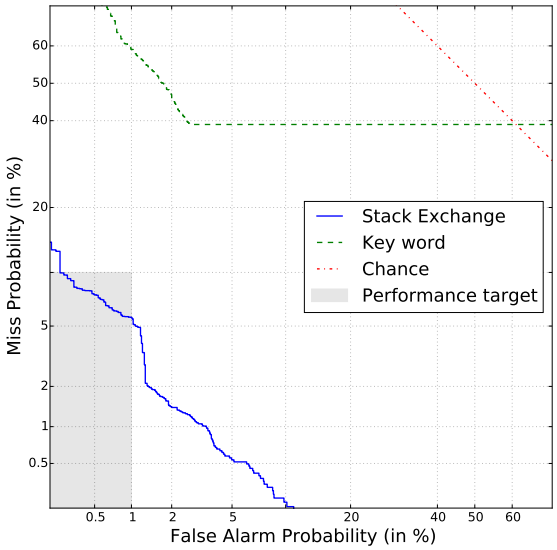

Figure 2: The DET curves for Stack Exchange documents show that the logistic regression classifier (blue curve) significantly outperforms both the baseline keyword system (green curve) and chance guessing (red curve).

cyber that are presented to an analyst. This means that, using the HLT classifier, an analyst has to examine 16.6 times fewer non-cyber documents to find the same number of cyber documents. In commonly-used HLT terminology we require recall (percentage of cyber documents labeled "cyber" by the classifier) above 90% and precision (percentage of cyber documents among all those labeled "cyber" by the classifier) above 83% for an input stream of documents containing only 5% cyber documents.

The curves shown in Figure 1 indicate that the classifiers we developed for each social media corpus do meet the performance target. They miss less than 10% of cyber-documents and classify less than 1% of the non-cyber documents as cyber. Before obtaining these results, we explored the effect of varying: the minimum number of words in each document, the amount of training data used, and types of text preprocessing. We also explored feature selection, using n-grams, and alternative classifiers. None of the exploratory results were substantially better than those shown here.

Although we focus on reaching the above performance target, there are times when analysts have time to examine only those few documents rated as having the highest probability of being cyber by the classifier. Performance in this case can also be determined by the DET curves shown. The simplest measure would be the precision (one minus the vertical miss probability) at the left of the curves where the false alarm probability is set as determined by the number of documents examined. In most cases, performance in the left correlates with performance in the target region so our discussion focuses on reaching the targeted gray box region in the rest of this paper.

4.1 Comparative Analysis of Classifiers

Figure 2 compares the performance of the baseline keyword classifier to the logistic regression classifier using Stack Exchange data. The logistic regression classifier (bottom blue curve) passes through the performance target region. The baseline keyword system (green curve) performs substantially worse. At a false alarm probability of 10%, the system fails to detect roughly 40% of the cyber documents; at a false alarm probability of 1%, the miss probability is roughly 60%. To determine the cause of this poor performance, we examined the Stack Exchange documents that corresponded with the false alarms. False alarms were often caused by one or more occurrences of cyber keywords in documents with topics unrelated to cyber. For example, the keyword "infected" appeared in documents referring to bacterial infection. Similarly, the keyword "checksum" appeared in many documents on technical topics. Simply counting occurrences of keywords without considering the context of the documents led to the false alarms. Worst-case performance, shown by the chance-guessing curve in the upper right (red), is obtained by randomly labeling each document as being cyber with a probability that ranges from zero to one.

Table 2 provides some insight into why our logistic regression classifier performs better than the keyword system. On the top are the 50 words that receive the highest positive weights and thus contribute more than other words in causing a document to be classified as cyber. These words span a wide range of cyber discussions on several topics. Many of these words and other positively weighted cyber words used by this classifier are highly likely to be present in cyber documents. Unlike the keyword system, our classifier strongly indicates cyber only if many of the 50 cyber words are combined in one document. Multiple instances of one word will not yield

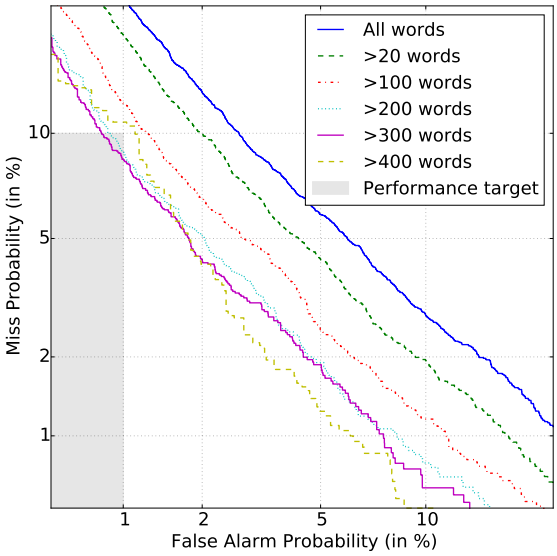

Figure 3: As the minimum number of words in each Reddit document is increased, the classifier's performance improves.

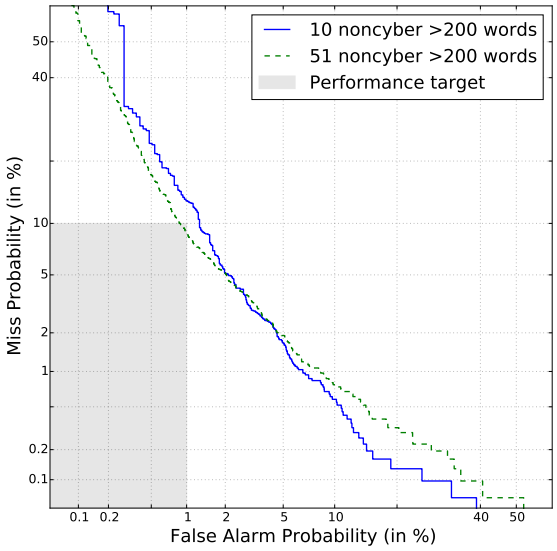

Figure 4: Decreasing the number of non-cyber sub-Reddit topics from 51 to 10 results in only a small degradation in performance but does move performance outside the target region.

a strong cyber indication. The bottom of this table lists the 50 words that receive the highest magnitude negative weights and thus contribute more than others in causing a document to be classified as non-cyber. These words indicate the breadth of topics that non-cyber documents cover. This diversity suggests that a large set of non-cyber documents needs to be fed into the logistic regression classifier to obtain good performance.

4.2 The Effect of Document Length and Amount of Training Data

The DET curves in Figure 3 show how performance depends on the number of words in a Reddit document. The upper blue curve shows performance using all 59k documents. The lower curves show performance only for documents that exceed the displayed word count. As seen, performance initially increases rapidly as the number of words increases. However, the rate of performance increase slows as the minimum number of words increases, and classifier performance enters the target range when the minimum number of words is above 200. Our results suggest that 200 or more words in an Internet conversation are required to provide accurate classification of cyber and non-cyber documents. In our application, this is a reasonable restriction on document length because analysts are looking for long-term cyber discussions and not isolated mentions of cyber topics.

The effect of reducing the amount of non-cyber Reddit training data was explored by training the above system with conversations from 10 instead of 51 non-cyber sub-Reddits. This reduces the total number of discussions from roughly 57k to 10k. As can be seen in Figure 4, this causes only a small decrement in performance.

Classifier performance also improves for Twitter as the number of words per document and the amount of non-cyber training data are increased (Figure 5). For Twitter, a document is composed of all the tweets from a single user, so the number of words per document is increased by including more tweets per user. The number of non-cyber training documents is increased by randomly sampling users and collecting their tweets in additional documents. Because we assume that there is a very low probability of a randomly sampled user discussing cyber topics, no extra labeling or cost is incurred by incorporating additional training data. The same 127 cyber users were used to obtain each of the results in Figure 5. On average, there are 10 words per tweet after preprocessing, so in each of the results with a minimum of 20 tweets, there are 200 words per document. Performance was further improved by collecting additional tweets and increasing the average number of words per document to 1,000. These results are consistent with the Reddit results showing improved classifier performance as more words are added to the documents and with the Reddit results showing improved classifier performance as more non-cyber training data are provided.

4.3 Stability in Performance over Time

Another test of our logistic regression approach determined whether a classifier trained before the Heartbleed vulnerability was made public could detect social media discussions concerning Heartbleed. Such discussions could only be found if they included words that were used in prior social network cyber discussions because the classifier would have never

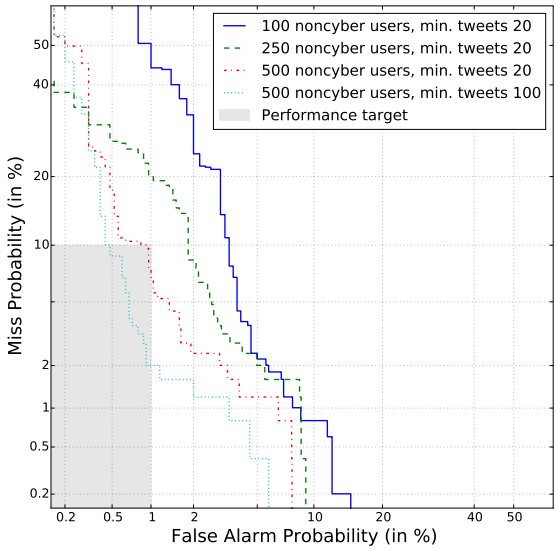

Figure 5: As the number of non-cyber users and tweets per user are increased, performance improves. For example, at 1% false alarms, the miss rate is 40% with 100 non-cyber users (dark blue curve), 20% with 250 users (green curve), and only 6% with 500 users (red curve). By adding additional tweets from the 500 users, the miss rate was reduced to 2% at 1% false alarms (light blue curve).

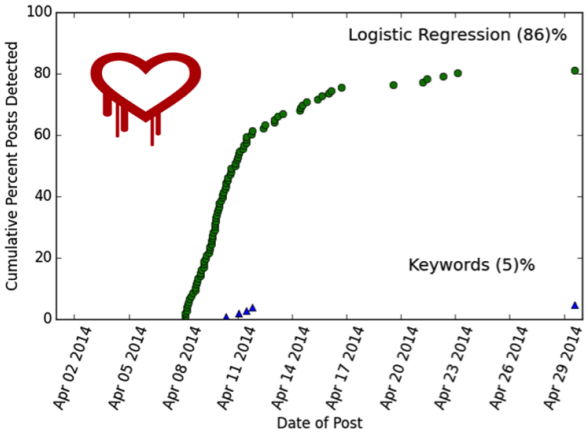

Figure 6: Our classifier detected 86% of the Stack Exchange posts discussing Heartbleed after training on historical data from before the Heartbleed vulnerability (green dots), in comparison the baseline keyword system detected only 5% of posts discussing Heartbleed (blue triangles).

seen the word "Heartbleed". Figure 6 plots the cumulative percentage of Stack Exchange threads detected by a logistic regression classifier trained on 3924 cyber and 7848 non-cyber documents posted before the Heartbleed attack was announced on 8 April 2014. The classifier immediately detects the flurry of posts on 8 April and in the following days. Of the 106 Heartbleed-tagged threads, 86% were detected and only 14% were missed at a false alarm rate of 1%. Our logistic regression classifier performed much better than the keyword baseline system, which only detected 5% of the Heartbleed discussions, because our system detects words related to the protocols affected by Heartbleed (e.g., SSL, TLS) and other words associated with cyber vulnerabilities (e.g., malware, overflow, attack). The keyword system lacked the keywords used in Heartbleed discussions, and thus suffered from a high miss rate.

A system to detect cyber documents is most useful if does not require frequent retraining to match possible changes in cyber vocabulary over time. We performed experiments in which a classifier was trained on Stack Exchange data up to a given date and then tested every month after that date without retraining. Figure 11 plots the miss percentage (averaged over false alarm rates ranging from 0.25% to 2.0%) for a classifier that was trained on data before June 2012 and then tested on the new data in each month for the following year. The results indicate that the miss rate increases little over the year and is always below 10%. The experiment was repeated over multiple time periods from 2012 through 2014, producing similar results each time. It is thus sufficient to retrain our classifier once per year or every six months.

5 Conclusions and Future Work

After developing three text corpora containing cyber and non-cyber documents we demonstrated that HLT classifiers perform well for all corpora providing a high concentration of cyber documents after filtering. Logistic regression classifiers provided good performance when there were more than roughly 200 words in a discussion. In one illustrative test, a classifier trained before the major Heartbleed vulnerability was announced could accurately detect discussions of this vulnerability and classifier performance was maintained without retraining even when tested on discussions occurring six months to a year after training. Classifiers developed can filter an input stream of documents containing only 5% cyber documents to a filtered set for an analyst containing from between 83% and 95% cyber documents.

Preliminary experiments suggest that performance degrades when a classifier is trained on one corpus (e.g., Reddit) and tested on another (e.g., Stack Exchange). To improve cross-domain performance, we are currently exploring using neural network word embeddings as word features to take advantage unlabeled training data (e.g. [Mikolov *et al.*, 2013]). We are also exploring adapting classifiers across domains and creating more general non-cyber word distribution models that can be used across corpora and created without document labels.

We have also begun collecting non-English social media content to extend our approaches to other languages. Follow-on work also includes automatically linking entities in discussions to cyber concepts as suggested in [Mulwad *et al.*, 2011]. One goal of this work will be to automatically fill in components of the "Diamond Model" of intrusion analysis [Calta-

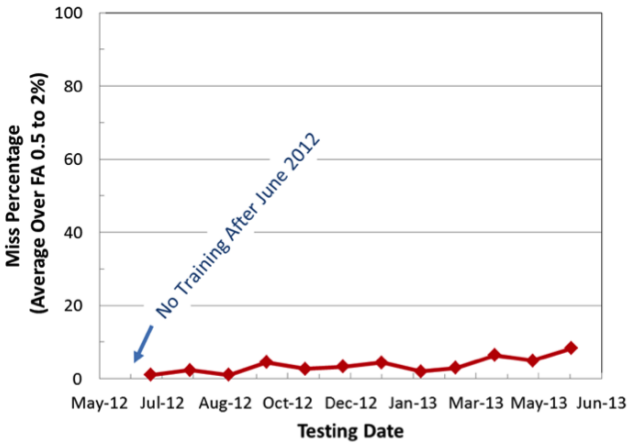

Figure 7: A classifier trained on Stack Exchange data before June 2012 and tested every month after that for one year stays within the performance target.

girone *et al.*, 2013] by extracting data concerning attacker capabilities, attacker infrastructure, and victims of cyber adversaries. A second goal is to improve performance with shorter cyber discussions containing fewer than 200 words and reduce the amount of training data required when extending our work to additional Internet social media forums.

Acknowledgements

We are grateful to Julie Dhanraj, Beth Richards, Jason Duncan, Peter Laird, and their subject-matter experts for their support. Vineet Mehta's initial contributions are appreciatively acknowledged. Thank you to Clifford Weinstein, Kevin Carter, and Kara Greenfield for initiating the program and providing helpful insights.

References

[Anderson, 2008] Ross J. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. Wiley, 2008.

[Auer *et al.*, 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. *Dbpedia: A nucleus for a web of open data*. Springer, 2007.

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[Caltagirone *et al.*, 2013] Sergio Caltagirone, Andrew Pendergast, and Christopher Betz. The Diamond Model of Intrusion Analysis. Technical report ada586960, Center for Cyber Threat Intelligence and Threat Research, Hanover, MD, 2013.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine learning*, 20(3):273–297, 1995.

[Hastie *et al.*, 2009] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, volume 2. Springer series in statistics Springer, Berlin, 2009.

[Jones *et al.*, 2015] Corinne L Jones, Robert A Bridges, Kelly Huffer, and John Goodall. Towards a relation extraction framework for cyber-security concepts. *arXiv preprint arXiv:1504.04317*, 2015.

[Jurafsky and Martin, 2008] Daniel Jurafsky and James Martin. *Speech and Langage Processing, 2nd Edition*. Prentice Hall, 2008.

[Lau *et al.*, 2014] Raymond Y.K. Lau, Yunqing Xia, and Yunming Ye. A Probabilistic Generative Model for Mining Cybercriminal Networks from Online Social Media. *IEEE Computational Intelligence Magazine*, 9(1):31–43, January 2014.

[Martin *et al.*, 1997] Alvin F. Martin, George R. Doddington, Terri M. Kamm, Mark L. Ordowski, and Mark A. Przybocki. The DET curve in assessment of detection task performance. In *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*, 1997.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[Mulwad *et al.*, 2011] Varish Mulwad, Wenjia Li, Anupam Joshi, Tim Finin, and Krishnamurthy Viswanathan. Extracting Information about Security Vulnerabilities from Web Text. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, volume 3, pages 257–260. IEEE, 2011.

[NIST, 2017] NIST. National Vulnerability Database (NVD). https://nvd.nist.gov/, 2017.

[Reddit, 2016] Reddit. https://www.reddit.com/, 2016.

[Sabottke *et al.*, 2015] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-Rorld Exploits. In *Proceedings of the 24th USENIX Conference on Security Symposium*, pages 1041–1056. USENIX Association, 2015.

[StackExchange, 2016] StackExchange. https://stackexchange.com/, 2016.

[Syed *et al.*, 2016] Zareen Syed, Ankur Padia, M. Lisa Mathews, Tim Finin, and Anupam Joshi. UCO: A Unified Cybersecurity Ontology. In *Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*. AAAI Press, February 2016.

[Twitter, 2016] Twitter. https://twitter.com/, 2016.